

Seminario JavaScript: Introduzione al linguaggio

Linguaggi per il Web

Ingegneria Informatica, Ingegneria dell'Informazione,
Sapienza Università di Roma, sede di Latina

25 novembre 2013

Dott. Corrado Di Benedetto

Argomenti

- Che cosa è JavaScript ?
- Cosa si può fare JavaScript ?
- Come e dove usarlo ?
- Librerie
- Cenni variabili e operatori
- Cenni strutture di controllo e cicli
- Funzioni
- Gestione degli errori
- Interazione con l'utente
- Eventi
- Validazione dati in una Form HTML

Che cosa è JavaScript ?

- Un linguaggio di scripting orientato agli oggetti usato comunemente nei siti web.
- Nel 1997 è stato standardizzato dalla organizzazione ECMA.
- Nel 1998 lo standard ECMA (ECMAScript-262) è diventato uno standard internazionale ISO (ISO/IEC 16262).

Cosa si può fare con JavaScript ?

1. Recuperare le informazioni del browser

```
function recuperaInfoBrowser()  
{  
    document.getElementById("Form1").Name.value=navigator.appName;  
    document.forms["Form1"]["Version"].value=navigator.appVersion;  
    document.forms["Form1"]["Code"].value=navigator.appCodeName;  
    document.getElementById("Form1").Agent.value=navigator.userAgent;  
}
```

Cosa si può fare con JavaScript ?

2. Creare Cookie

```
function impostaCookie (nome, valore, scadenza)
{
  if (scadenza == "") {
    var oggi = new Date();
    oggi.setMonth(oggi.getMonth() + 3);
    //restituisce la data nel formato necessario
    scadenza = oggi.toGMTString(); }
  valore = escape(valore);
  document.cookie = nome + "=" + valore + "; expires = " +
scadenza;
}
```

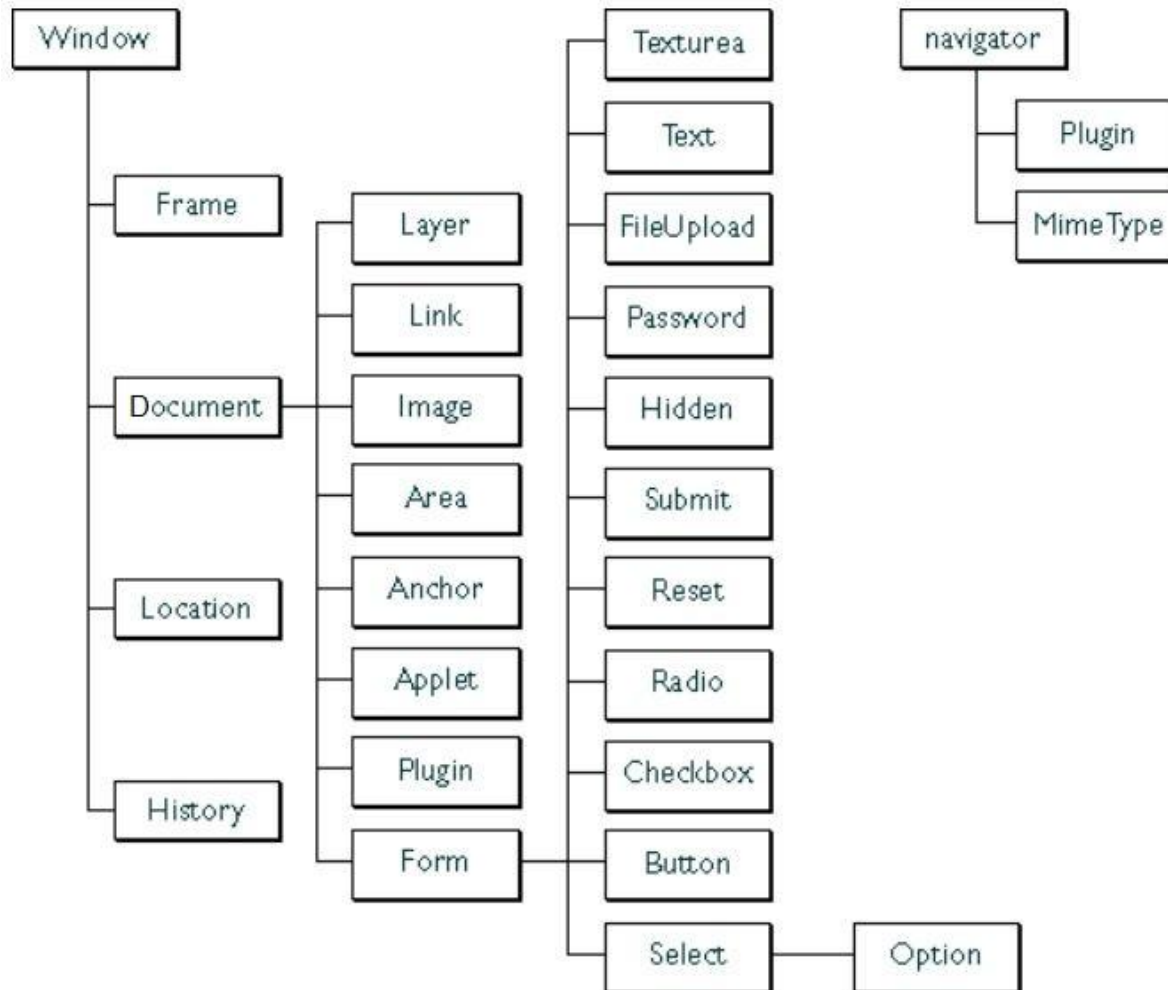
Cosa si può fare con JavaScript ?

3. Manipolare il DOM

```
function impostabgColor (value)
{
    document.bgColor = value ;
}
```

Il DOM rappresenta il modello a oggetti di una pagina web o di un documento HTML e possiede metodi per navigare e modificare la sua struttura (creare, eliminare, inserire, spostare, scambiare, sostituire oggetti).

Document Object Model



Cosa si può fare ?

4. Gestione degli eventi

```
function click()  
{  
    alert ("Ciao!") ;  
}
```

```
<input type="button" name="Benvenuto" onClick="click()" >
```

L'applicazione deve gestire gli eventi scatenati dall'utente o dal browser, come ad esempio il clic su un oggetto (es. Button).

Gli oggetti hanno dei gestori di eventi (es. onClick) attraverso i quali è possibile eseguire comandi JavaScript o chiamata a funzioni personalizzate.

Cosa si può fare con JavaScript ?

5. Validare i dati in una form HTML

```
function validaUserForm()  
{  
    var x=document.forms["userForm"]["userName"].value;  
    if (x==null || x=="")  
    {  
        alert("Devi inserire un nome.");  
        return false;  
    }  
}
```

Come usarlo ?

- Per inserire un codice JavaScript in una pagina HTML, è necessario usare `<script>`.
- L'attributo `type` specifica il tipo di script usato: `application/x-javascript` oppure `text/javascript` (non standard), è possibile ometterlo.
- Sono stati deprecati:
 1. l'attributo `language` al posto di `type` (`language="JavaScript"`);
 2. il codice tra delimitatori di commento
`<!-- ...codice... // -->`.

```
<script type="application/x-javascript">  
    codice...  
</script>
```

Dove usarlo ?

- Il codice JavaScript può essere inserito all'interno nella sezione **head** o nella sezione **body** della pagina **HTML** .
- Le istruzioni vengono eseguite in ordine sequenziale (eccezione eventi/funzioni) e la differenza è che il codice dentro head viene eseguito prima del caricamento completo della pagina.

```
<html>
  <head>
    <title>...</title>
    <script>...</script>
  </head>
  <body>
    <script>...</script>
  </body>
</html>
```

Librerie

- È possibile usare l'attributo `src` per inserire il path di una libreria JavaScript

```
<script type="text/javascript" src="..."></script>
```

locale

```
src="Script/jquery-1.7.1.min.js"
```

```
src="Script/myLib.js"
```

in rete

```
src="http://code.jquery.com/jquery-1.7.1.min.js"
```

```
src="http://code.jquery.com/jquery-1.7.1.js"
```

Le variabili

- Una variabile identifica una cella di memoria nella quale vengono salvati i dati durante l'esecuzione dello script. JavaScript è un linguaggio a **debolmente tipizzato**, è facoltativo indicare il tipo della variabile.
- Per usare una variabile è necessario indicare al parser il suo nome utilizzando l'istruzione `var`:

```
var nomeVariabile;
```

La variabile non ha valore fino a quando non viene inizializzata.

Le variabili

- E' possibile **inizializzare** la variabile contestualmente alla dichiarazione:

```
var nomeVariabile = espressione.
```

Per `espressione` si intende una sequenza di operatori, variabili e/o dati che restituisca a sua volta un valore.

- L'operazione fondamentale delle variabili è **l'assegnazione**, attribuzione di un valore ad una variabile:

```
var nomeVariabile;           // definizione
nomeVariabile = valore;     // assegnazione
```

dove `valore` è un'espressione.

Esempi

- `var1 = "ciao!";`
- `var2 = 3;`
- `var3 = false;`
- `var4 = var3; //var4 copia del valore di var3`

Operatori matematici

Operatori	Descrizione	Esempi	Valore	Risultati
+	Addizione	$x=y+2$	$y=5$	$x=7$
-	Sottrazione	$x=y-2$	$y=5$	$x=3$
*	Moltiplicazione	$x=y*2$	$y=5$	$x=10$
/	Divisione	$x=y/2$	$y=5$	$x=2.5$
%	Modulo	$x=y\%2$	$y=5$	$x=1$
++	Incrementa di uno	$x=++y$	$y=6$	$x=6$
		$x=y++$	$y=6$	$x=5$
--	Decrementa di uno	$x=--y$	$y=4$	$x=4$
		$x=y--$	$y=4$	$x=5$

Operatori stringa

- La concatenazione è l'unione di due stringhe.
- In JavaScript l'operatore di concatenazioni è +.

- **Esempi:**

```
var nome = "Mario";  
var cognome = "Rossi";
```

```
var messaggio = " Mi chiamo " + nome + " "+ cognome  
                + ".";
```

```
// messaggio: Mi chiamo Mario Rossi.
```

Operatori booleani

Sono quelli che restituiscono un valore booleano (vero o falso) e confrontano due valori dello stesso tipo.

Operatori	Descrizione	Esempi	Risultati
<	minore	$3 < 4$	true
<=	minore o uguale	$3 <= 3$	true
==	uguale	$4 == 5$	false
>=	maggiore o uguale	$5 <= 9$	true
>	maggiore	$3 > 4$	true
!=	diverso	$5 != 5$	false

Operatori logici

È possibile comporre le espressioni booleane usando gli operatori logici, che lavorano con valori booleani e restituiscono a loro volta un valore booleano.

Operatori	Descrizione	Esempi	Risultati	Valore
&&	congiunzione logica	$(a > 0) \ \&\& \ (a < 100)$	true	devono essere vere le espressioni a destra e sinistra dell'uguale.
	disgiunzione logica	$(a < 0) \ \ (b > 20)$	true	se almeno una delle due espressioni è vera
!	negazione logica	$!(a < 0)$	true	se il valore di a non è minore di 0

Strutture di controllo

```
if(condizione){istruzioni;}
```

```
if (condizione)
{
    istruzioni;
}
else
{
    istruzioni;
}
```

```
if (condizione1)
{
    istruzioni;
}
else if (condizione2)
{
    istruzioni;
}
else
{
    istruzioni;
}
```

```
switch (espressione)
{
    case valore1:
        istruzioni;
        break;
    case valore2:
        istruzioni;
        break;
    default :
        istruzioni;
}
```

Cicli

```
for (variabile in oggetto)
{
    istruzioni;
}
```

```
while(condizione)
{
    istruzioni;
}

do
{
    istruzioni;
}
while (condizione)
```

```
for ([iniziale]; [condizione]; [incremento])
{
    istruzioni;
}
```

Funzioni

- Una funzione è un blocco di istruzioni identificato da una nome che può accettare o meno parametri e può restituire un valore.
- La sintassi per la creazione di una funzione è la seguente:

```
function nomeFunzione (arg1, arg2,..., argN)
{
    // Codice
    // return ...; se restituisce un valore
}
```

Esempi di funzioni

Definizione:

```
function visualizzaMessaggio()  
{  
    alert("Benvenuto!");  
}
```

```
function visualizzaMessaggio(nome)  
{  
    alert("Benvenuto, "+nome+"!");  
}
```

```
function sommaDueNumeri(a, b)  
{  
    var somma = a + b;  
    return somma;  
}
```

Utilizzo:

```
visualizzaMessaggio();  
// "Benvenuto!"
```

```
visualizzaMessaggio("Mario!");  
// "Benvenuto Mario!"
```

```
var c = sommaDueNumeri(3,5);  
//c assume valore 8
```

Gestione degli errori

- Uno script può terminare normalmente o “sollevare un’eccezione”.
- Un’eccezione è un oggetto restituito che descrive una situazione anomala o di errore.
- Un’eccezione viene lanciata in un punto del codice ed utilizzando il costrutto `try...catch...finally` può essere “catturata” e gestita.

Gestione degli errori

Dettaglio sintassi:

```
try {  
    // Istruzioni in cui possono essere lanciate  
    // delle eccezioni  
}  
catch (error) {  
    // Istruzioni da eseguire in caso di eccezione  
}  
finally {  
    // Istruzioni da eseguire successivamente  
    // in entrambi i casi  
}
```

Gestione degli errori

- Inizialmente, vengono eseguite le istruzioni all'interno del blocco `try`.
- Se viene lanciata un'eccezione, l'esecuzione salta immediatamente alle istruzioni del blocco `catch` e l'eccezione viene resa disponibile come argomento `error`.
- In caso contrario, il blocco `catch` viene saltato.
- Una volta che un dei blocchi si è concluso vengono eseguite sempre le istruzioni nel blocco `finally`.

Gestione degli errori

- L'istruzione `throw` consente di creare un'eccezione, e se usata in un blocco `try ... catch` è possibile controllare il flusso del programma.
- L'eccezione può essere una stringa, un numero, un booleano o un oggetto.
- La sintassi è la seguente:

```
throw eccezione;
```

L'interazione con l'utente

- Uno script all'interno di una pagina web può interagire con l'utente nei seguenti modi:
 1. Finestra di avviso;
 2. Finestra di conferma;
 3. Finestra di richiesta input;
 4. Standard output.

Esempi interazione utente

```
1. function mostraAvviso()  
  {  
    alert("Ciao!");  
  }
```

```
2. function mostraConferma()  
  {  
    var r = confirm("Inviare la mail?");  
    if (r == true)  
    {  
      alert("Mail inviata!");  
    }  
    else  
    {  
      alert("Invio mail annullato!");  
    }  
  }
```

Esempi interazione utente

```
3. function mostraRichiestaInput()
{
    var name = prompt("Please enter your name", "Mario");
    if (name != null && name != "")
    {
        document.write("Ciao " + name + "!");
    }
}
```

```
4. function mostraStandardOutput()
{
    document.write("<h3>Ciao " + name + "!</h3>");
}
```

Eventi

- Gli elementi HTML possono generare degli eventi ai quali può corrispondere un'azione, se è stato registrato un gestore di eventi. Questi gestori sono spesso delle funzioni definite all'interno del HTML.
- La sintassi per richiamare un evento in uno script è la seguente:

```
Oggetto.evento = handler;
```

Esempio eventi

```
<html>
  <head>
    <script type="text/javascript">
      function mostraAvviso()
      {
        alert("Ciao!");
      }
    </script>
  </head>
  <body>
    <input type="button"
      onclick="mostraAvviso()"
      value="avviso" />
  </body>
</html>
```

http://www.w3schools.com/jsref/dom_obj_event.asp

La validazione dei dati in una form HTML

- JavaScript può essere usato per validare i dati in una form HTML prima dell'invio al server.
- I controlli più comuni sono relativi a:
 1. campi obbligatori
 2. campi data
 3. campi numerici
 4. validità formale di un pattern (e-mail, codice fiscale,...)

Esempio validazione form

```
<html>
  <head>
    <script type="text/javascript">
      function validaUserForm ()
      {var x=document.forms["userForm"]["userName"].value;
        if (x==null || x=="")
          {alert("Devi inserire un nome."); return false;}}
    </script>
  </head>
  <body>
    <form name="userForm" action= "userForm.php" onsubmit="return
      validaForm()" method="post">
      Nome: <input type="text" name="userName">
      <input type="submit" value="Invia">
    </form>
  </body>
</html>
```

Esercizi

Esercizio 1

Creare una pagina HTML contenente un elenco di link.

Quando l'utente passa sopra (onmouseover) un link mostrare un messaggio sulla barra di stato. Il messaggio deve essere diverso per ogni link.

Esercizio 2

Creare una funzione JavaScript che cambi il colore di sfondo (document.bgcolor) ogni due secondi.

La funzione deve essere mandata in esecuzione dopo il caricamento della pagina (onload).

Esercizio 3

Modificare l'esercizio precedente in modo che il cambio di colore di sfondo viene attivato con la pressione (onclick) di un bottone.

Inserire un secondo bottone per interrompere l'animazione.

Esercizi

Esercizio 4

Creare uno script che chieda il nome e il cognome all'utente prima che la pagina venga caricata.

Provare a chiamare una finestra di dialogo che contenga la scritta "Benvenuto"+ nome + cognome.

In fondo alla pagina deve essere scritto "Benvenuto"+ nome + cognome.

Esercizio 5

Creare una pagina che chieda all'utente di fornire la propria autorizzazione al trattamento dei dati.

Nel caso non sia data, scrivere "Autorizzazione non concessa".

Nel caso contrario scrivere "Autorizzazione concessa".

Esercizio 6

Scrivere uno script che stampi il fattoriale di un numero ricevuto in input (metodo prompt) dall'utente.

Lo script deve verificare che il numero si compreso tra 1 e 15, in caso contrario deve segnalare l'errore all'utente.

Esercizi

Esercizio 7

Scrivere uno script che stampi la tabellina di un numero, compreso tra 1 e 10, ricevuto in input dall'utente.

Esercizio 8

Chiedere tre colori in input all'utente: uno per il testo, uno per lo sfondo e uno per il titolo.

Chiedere all'utente la dimensione del carattere di base (da 1 a 7).

Determinare la dimensione e le caratteristiche di H1 (+3, grassetto), H2 (+2) e H3 (+1, grassetto, corsivo) rispetto alle caratteristiche inserite dall'utente per il carattere.

Creare un testo in cui siano evidenti queste caratteristiche.

Controllare che il colore dello sfondo, quello del titolo e quello del testo siano diversi.

Nel caso due o più dei colori assegnati dall'utente siano uguali, devono essere presenti regole che determinino che colore devono assumere titolo, testo e sfondo.

Esercizi

Esercizio 9

Scrivere uno script che chieda una serie di numeri all'utente. Si interrompa quando l'utente inserisce al posto di un numero la parola chiave "end". Interagisce con l'utente nel caso abbia inserito dei valori non validi. Chieda conferma prima di calcolare la media dei numeri introdotti. Stampi a video il risultato in una tabella.

Esercizio 10

Implementare la funzione `celsius` che restituisce l'equivalente Celsius di una temperatura Fahrenheit usando la formula

$$C = 5.0/9.0 * (F - 32)$$

Implementare la funzione `fahrenheit` che restituisce l'equivalente Fahrenheit di una temperatura Celsius usando la formula

$$F = 9.0/5.0 * C + 32$$

Le funzioni devono verificare se il parametro immesso è di tipo numerico. Testare le funzioni inserendo le loro definizioni in un documento HTML contenente uno script che chiede all'utente di immettere un valore e visualizza il risultato dell'invocazione della funzione sul valore immesso.

Risorse

JavaScript

- <http://it.wikibooks.org/wiki/JavaScript/Introduzione>
- <http://it.wikipedia.org/wiki/JavaScript>
- <http://www.w3schools.com/js/default.asp>
- <http://www.w3schools.com/jsref/default.asp>
- <http://www.ecma-international.org/publications/standards/Ecma-262.htm>;

DOM

- <http://www.w3schools.com/html/dom/default.asp>

jQuery

- <http://jquery.com/>
- http://docs.jquery.com/Downloading_jQuery
- <http://www.w3schools.com/jquery/default.asp>